The Future of Wires

Mark Horowitz, Ron Ho, Ken Mai
Stanford University, California
{horowitz,ronho,demon}@vlsi.stanford.edu

1.0 Introduction

At first glance, the future of wires in integrated circuit technologies looks grim. Even projections with copper interconnections and low-κ dielectrics show that wire delay for a fixed length wire will increase at a rate that is greater than linear with scaling factor [1]. This has led to a number of papers which have predicted the demise of conventional wires, and the need for new interconnection methods.

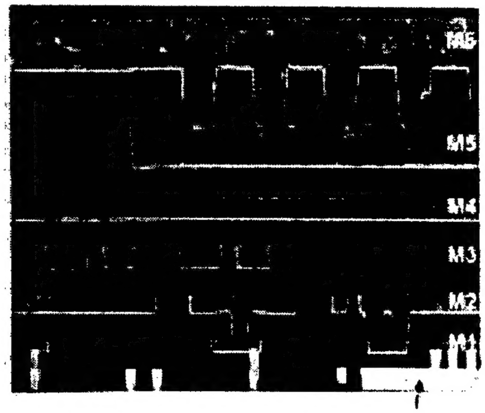
This chapter examines wire scaling and the capabilities of future wiring systems in more detail to try to better understand the constraints of these systems. The results are a little surprising. If an existing circuit is scaled to a new technology, the relative change in the speed of wires versus the speed of gates is modest. Depending on your assumptions on transistor performance under scaling, low-κ dielectrics, and higher aspect ratio wires, the ratio is close to one. Thus the performance of today's IP cores should continue to improve with technology scaling. The key part of this scenario is the length of the wire measured in gate pitches has remained constant, so its length in microns has scaled. The problem that designers face is not in needing to repartition current designs into smaller blocks, but rather that wire performance does not improve fast enough to make global communication on wires in a billion-transistor chip is free. This global communication will be cheaper than in today's large board-level systems, but it still will take multiple cycles to send data across the chip. This constraint has already influenced some current designs and will lead to new architectures, that are still wire-based, with partitioned resources and more explicit communication mechanisms.

In this paper we first discuss the wire metrics of interest and examine them in a contemporary 0.25μm process. We then discuss technology scaling over the next several generations, from SIA and other predictions, and how our wire metrics trend over that time. We will examine the delay and bandwidth limitations of both long global wires and short local wires and discuss architectural design techniques that help us avoid the limitations of scaled wires.

2.0 Wire metrics

To ground our discussion of wire metrics, we will use a contemporary 0.25μm baseline technology, which has five to six layers of aluminum or copper interconnect, with upper metals wider and taller than lower metals. The lowest level, M1, has the finest pitch and hence the highest resistivity, and typically connects nets within gates or between relatively close gates. The middle layers, M2 through M4, have a wider pitch than M1 and are used for both short and long-haul routes. The top layers, M5 and M6, have the widest pitch and thus the lowest resistivity and typically carry global routes, power, ground, and global clock. The following figure gives typical pitches for these various layers in technology-independent λs, where 1λ is half the drawn gate length.

FIGURE 1. A 0.25μm technology with typical pitches



Typical metal pitches

Layer	Width and spacing
M5-6	8 λ and 8 λ
M2-4	4 λ and 4 λ
M1	3 λ and 2 λ

Picture source: N. Rohrer, ISSCC 1998

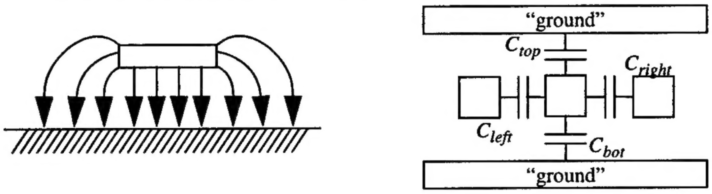
2.1 Resistance and capacitance

Aluminum wires typically have a resistivity of 3.3 mΩ*cm, while copper wires a resistivity of 2.2 mΩ*cm. Because of a thin barrier layer to prevent the copper from diffusing into the neighboring oxide, a wire's resistance as we migrate from Al to Cu does not quite decrease by 50%, although it does drop significantly. Designers can use the lower resistance of copper in three principal ways: they can keep the same wire pitch and thickness and thus benefit from lower resistance; they can maintain the same wire pitch and resistance by cutting down the thickness and thus benefit from lower cross-coupling and total capacitance; or they can maintain the same resistance and mostly the same capacitance by cutting down the pitch and thus benefit from much denser interconnect. In all three cases, designers benefit from copper's improved electromigration characteristics. In practice, designers have generally opted for the second technique since cross-coupled noise presents a difficult problem for large and high-performance designs [2].

Interconnections between metal layers (plugs or vias) for aluminum wires are made of tungsten and tend to be fairly resistive; in a 0.25μm process a M1-M2 via resistance is about 5Ω, and vias from M5 down to the substrate add up to more than 20Ω. This may seem large since a 1μm wide and 1mm long M5 line itself has a total wire resistance of only 20Ω, but designers can simply array many vias together; besides, electromigration constraints often demand multiple vias. The drawback to these via arrays is that they create large routing obstructions on the intervening M1-M4 metal layers. Copper processes improve on this situation by pouring vias out of copper at the same time the wires are deposited. These copper vias are much less resistive and do not need to be as aggressively arrayed.

Many models exist for the capacitance of a single isolated wire running over a ground plane; the better ones combine a bottom-plate area term with a fringing term to account for field lines emerging from the edge and/or top of the wire. However, wires today are as tall as they are wide and will grow even taller to decrease resistance, so their side-to-side capacitances are a significant portion of the total. Their capacitance is therefore more conveniently modeled by four parallel-plate capacitors for the top, bottom, left, and right sides [3], plus a constant term for fringing capacitance. Capacitors on the top and bottom plates refer to ground, since they are to orthogonally-routed wires that, averaged over the length of the wire, tend to maintain a constant voltage (this is not true if the orthogonal wires switch simultaneously and monotonically, like in a precharge bus). However, capacitors to the left and right have data-dependent effective capacitances that can vary delay. The left and right neighbors are also the worst offenders for noise injection. For the very top layers of metal with no upper layers, we use three parallel plates with fringing terms on the two sideways capacitors.

FIGURE 2. Isolated and realistic on-chip capacitance models



2.2 Delays and bandwidth

We calculate wire delays using a simple RC product. The delay of a gate driving a wire is

$$D = R_{gate}(C_{diff} + C_{wire} + C_{load}) + R_{wire}(C_{wire}/2 + C_{load}) \qquad \text{(EQ 1)}$$

This is only an approximation, since it ignores input slew rates: if a preceding wire is long enough that its end voltage slews very slowly, it will degrade the delay of the next gate. To achieve a measure of process independence, we will normalize all of our delay calculations to the delay of an inverter driving a fanout of 4. In a 0.25μm technology this delay is about 90pS. Any CMOS gate delay is roughly the same number of FO4s over a wide range of technology, process corner, temperature, and voltage. Wire delays do not scale with FO4 delays as well as other gates do, but they still match reasonably well.

The first term in the delay equation is 1FO4 since simple sizing heuristics aim for gate sizes to have a fanout of about 4; such sizing rules avoid huge gates for really long wires by upping the fanout or by considering the resistive shielding of downstream capacitance. We will also assume that $C_{wire} >> C_{load}$ for the long wires considered in this paper, since 1μm of transistor gate has capacitance equivalent to 10μm of interconnect wire, and we are considering wires in excess of 1000μm. Our metric for delay is therefore simply $1FO4+R_{wire}C_{wire}/2$. Our assumptions break down for wires driving large or many gate loads, such as control lines driving each bit of a 64-b datapath. The resistance, capacitance, and variable portion of delay for a minimum pitch unbuffered wire is below.

0.25μm tech	**M1**	**M2-M4**	**M5-M6**
Width, μm	0.375	0.5	1.0
Spacing, μm	0.25	0.5	1.0
Height, μm (1.8 aspect ratio)	0.675	0.9	1.8
Resistance (Al), Ω/mm	130	73	18
Resistance (Cu), Ω/mm	108	57	13
Capacitance, fF/mm (ε_r=3.9)	296	230	230
% of Cap is Xcap	78%	69%	69%
Wire delay (Al), FO4/mm^2	0.21	0.09	0.02
Wire delay (Cu), FO4/mm^2	0.18	0.07	0.01

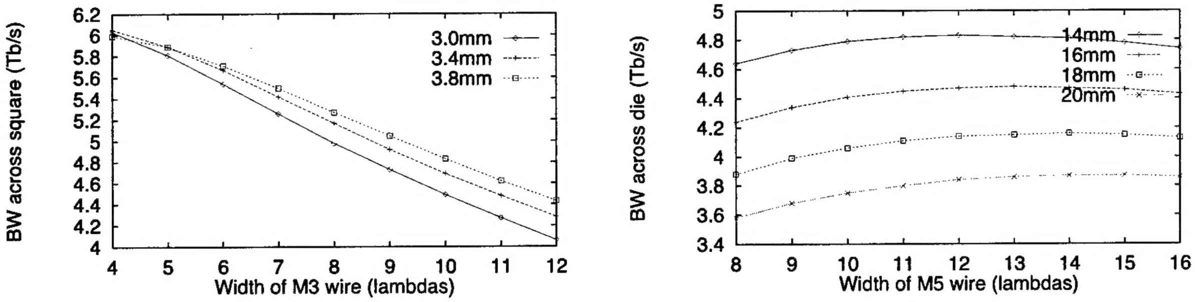
The table implies that today, long unbuffered wires with small loads are not too slow. On copper, a 1cm route takes 1+18=19 FO4s on M1, but 1+7=8 FO4s on M2-4 and only 1+1=2 FO4s on M6. This analysis ignores the effects of significant gate loads, which cause delays to grow rapidly.

We can estimate the bandwidth of an unbuffered wire by asking how long we must wait between successive transitions on a wire. If we switch a wire once, we need to wait until residual currents from that transition have mostly died away, or else we will see intersymbol interference when we switch it again. We can do this by waiting three propagation delays before sending the next signal, which gives us enough time for the output to transition past 90%. More aggressive techniques can achieve higher bandwidth; this estimate is only a first cut for comparative purposes. In the equation below we are assuming the propagation delay is a gate delay (1FO4) plus the distributed wire delay ($0.4R_wC_w$).

$$BW_{area} = \frac{1}{3FO4 + 1.2R_{wire}C_{wire}} \cdot \frac{areawidth}{wirepitch} \qquad \text{(EQ 2)}$$

This formulation allows us to examine interconnect bandwidth in both global and local contexts. In a global context, we run top level metals (M5 and M6) across the entire die and thus consider the bandwidth across a die-sized square. For local contexts, we can run middle layer metals (M2 to M4) across a synthesized functional block, a much smaller square whose size is set by the number of gates, the gate pitch, and cell utilization. In a 0.25μm technology, a 50K-gate design (today, synthesis is limited to 30K- to 100K-gate blocks) takes a square roughly 3.5mm on a side.

FIGURE 3. BW of unrepeated M3 and M5 lines across functional blocks of varying edge lengths

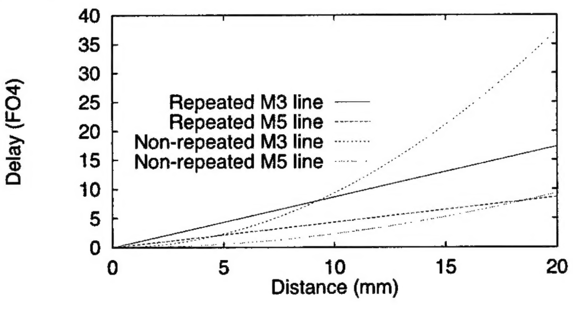


Not surprisingly, in a local (M3) context the bandwidth drops as we use wider and wider wires; the resistance improvement for wider wires is less than the area taken up by the wider wires. However, as seen above in Figure 3, for the M5 wires that run across the die, the quadratic dependence of delay on length becomes important, and making wires wider is worth the loss in routing density. Also not surprisingly, as the block or die edge lengthens, the BW decreases since it takes longer and longer to send tokens down a wire.

2.2.1 Repeaters

Inserting repeaters (either inverters or buffers) periodically along a wire breaks the quadratic dependence of wire delay on the wire length [4]. When repeaters divide a long wire into multiple shorter segments, the total wire delay is the number of segments multiplied by the individual segment delay. Each segment's delay is still quadratically dependent on its segment length, but the total wire delay is now linear with total length. Using a simple Elmore delay model for the repeaters (inverters in this case) and wire, we can derive the optimal segment length and repeater size for minimum total delay. The derivation is simple and yields the optimal segment length that makes the wire delay equal to the buffer delay. The results are shown below. Again, we note that this calculation does not consider significant gate loads: this figure implies that repeaters are never necessary in a 0.25μm technology. If gate loads are included, however, the intersection points between repeated and non-repeated graphs moves quickly to the left.

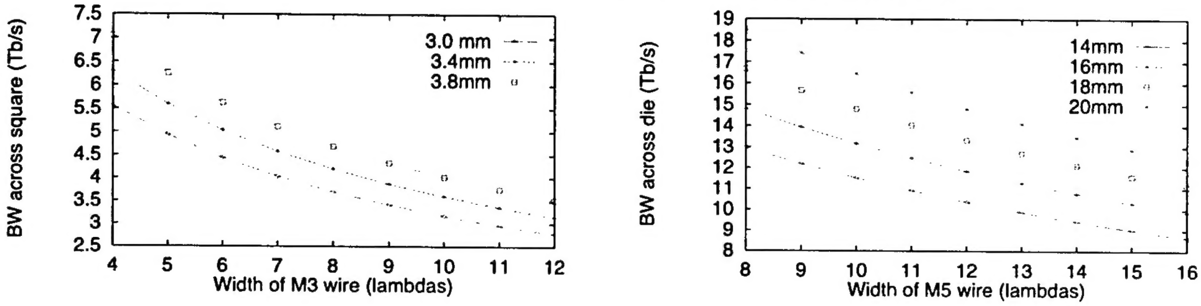
FIGURE 4. Repeated and non-repeated M3 and M5 lines for a 0.25μm technology



Insertion of repeaters into a design is not trivial. First, using inverters as the repeater element requires an even number of repeaters along the wire to avoid a logic inversion on the net. Second, repeaters are rarely placed in their optimal locations, since they require active area on the substrate; for floorplanning purposes repeaters are usually clustered in predefined locations. Third, repeating global wires requires many via cuts from the top level metal down to the substrate, potentially blocking out routes on intervening metal layers. Finally, the optimal repeater formulation requires very large devices, and if they are used to repeat a wide bus, the total device area can be large and perhaps push out the bus pitch. Fortunately, the delay curves of repeated wires have fairly shallow optimal points, so the costs of adding or removing an extra inverting repeater, of moving repeaters in the floorplan, or of reducing the size of the repeaters are not large.

The bandwidth of a repeated wire is much higher than that of a non-repeated wire. After sending one signal down a wire, we only need wait until that signal fully transitions on the first repeater segment before we send the next signal, so the bandwidth of a repeated wire does not depend on the full wire length. Also, since with optimally repeated wires, the delay of a single repeater segment is independent of wire width, wider wires always decrease bandwidth since they only reduce the number of signals that fit in a box. The bandwidth considering repeaters is shown below.

FIGURE 5. Bandwidth of repeated M3 and M5 unloaded lines



2.3 Power dissipation and coupling

In addition to delay and bandwidth metrics, we also consider the power and noise characteristics of wires. The power a wire dissipates is $P=CV\Delta Vf'$, where C is the line capacitance, ΔV the voltage swing, V the power supply, and f' the effective switching frequency. Strategies for reducing power dissipation aim to reduce both the voltage swing and power supply and thus gaining a V^2 decrease in power, and to encode the data in patterns that lower the effective switching frequency.

From Figure 2 we see that cross-capacitance to sideways neighbors is a significant portion of the total capacitance; because of the high aspect ratio, for M3 and M5 layers it is almost 70% of the total. Because sideways neighbors are not "grounded," they can inject coupled noise and increase or decrease delay. If the wire in the middle goes a different direction from its two simultaneously-switching neighbors, the effective sideways capacitance doubles. If the wire in the middle pulls in the same direction as its neighbors, though, the effective sideways capacitance is zero. For our 0.25μm technology, the effect on delay can vary from +40% to -70%, causing both max-delay and min-delay problems.

Injected noise from coupling capacitance can cause glitches and even part failures. The common model of $V_{peak} = \Delta V*C_{coup}/C_{tot}$ is often used for peak injected noise, but designers recognize it as too pessimistic for realistic designs, since it assumes the victim node is undriven and the attacker is infinitely driven. A better model for peak noise would include the time constants of the attacker and victim, and one convenient model is:

$$V_{peak} = \Delta V \cdot \frac{C_{coup}}{C_{tot}} \cdot \frac{1}{1 + \tau_{att}/\tau_{vic}} \qquad \text{(EQ 3)}$$

where τ_{att} and τ_{vic} are the time constants of the attacker and victim, respectively. If the attacker has a much smaller time constant than the victim (and is hence much stronger), the model approaches the pessimistic worst-case equation. This model is provably conservative for simple circuits with no wire resistance, and for circuits with wire resistance it is conservative for all simulations covering a wide range of interconnect parameters.

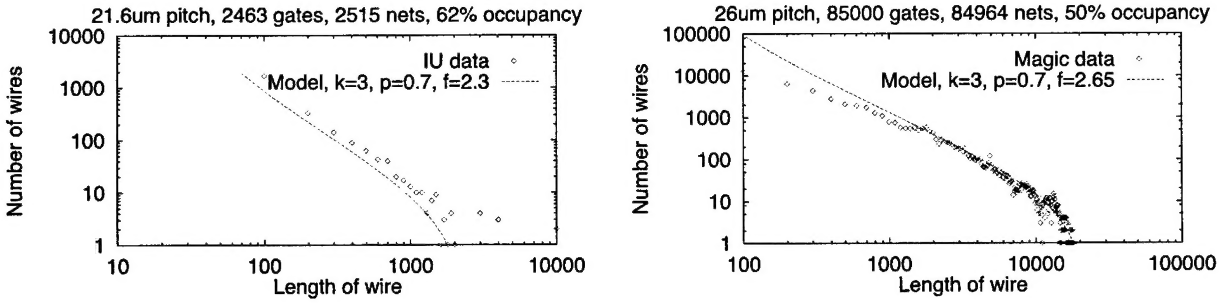
A future source of noise is inductive coupling from wide buses that simultaneously switch. Since magnetic fields extend much farther than just a nearest neighbor, a victim wire surrounded by ten or twenty attackers that all switch down can be coupled up [5]. To cope with these noise issues, designers work in a constrained space which guarantee the worst-case noise situations can not arise.

Delay and capacitive noise problems decrease with less cross-capacitance, and inductive noise decreases with better return paths; but in all three cases the principal design problem is one of extraction and then data management for the astounding number of extracted capacitors, resistors, and inductors generated by large designs.

2.3.1 Wire length statistics

Davis *et al.* constructed a model for wire length distributions, given Rent parameters and gate count [6], and showed it was a good fit for many designs, including microprocessors. We applied their model to four specifically synthesized blocks: three units from the M-Machine, a fine-grained multicomputer designed at MIT and Stanford [7], and the global placement of Magic, a protocol processor chip from Stanford's Flash multiprocessor [8] (minus the artificially long hand-routed MiscBus). From Figure 6 (we only show one M-Machine plot for brevity), we see that the model is a good fit for the wire length distributions of these designs, which span a wide range of gate count. The outliers in the M-Machine data are from long buses.

FIGURE 6. M-Machine and Magic/Flash wire load models



21.6um pitch, 2463 gates, 2515 nets, 62% occupancy

26um pitch, 85000 gates, 84964 nets, 50% occupancy

Since these graphs are on a log-scale plot, we can see that the vast majority of wires in a block are fairly short. If we define a "long" wire as a wire whose intrinsic RC delay exceeds a fraction of a gate delay, then we can see that the number of long wires is fairly small, under a few percent of the wires. This implies that wire delays for almost all of the wires are small, and that lower, high-resistance layers are adequate for those routes.

3.0 Technology scaling

Using technology projections like the NTRS SIA roadmap, we can predict the pitches of local, intermediate, and global interconnect [9]. The SIA projections for M1 pitch are close to 5λ, and we will assume that intermediate and global layers still use 8λ and 16λ pitches. We scale typical-typical FO4 delays in pS using the heuristic of $FO4 = 360 \cdot L_{drawn}$ (where L_{drawn} is in μm) and consider that high-performance chips will have cycle times of 16-20 FO4s.

L_{drawn}	**0.25μm**	**0.18μm**	**0.13μm**	**0.10μm**	**0.07μm**	**0.05μm**
Mid-layer width (4λ) in mμ	0.50	0.36	0.26	0.20	0.14	0.10
Global layer width (8λ) in μm	1.0	0.72	0.52	0.40	0.28	0.20
Chip edge length, mm	17.3	19	20.7	22.8	24.9	27.4
FO4 delay, pS	90	65	48	36	25	18
Frequency at 16 FO4s, GHz	0.7	1	1.3	1.7	2.5	3.5

Other parameters are also predicted by the SIA roadmap. Since predicting future technology scaling is always a difficult proposition, we will adopt two contrasting technology scalings: the first will be a conservative scaling (similar to Sylvester [10]) that does not assume some technological improvements: specifically, that low-κ relative dielectrics will scale at only 0.9x per technology and hence bottom out around ε_r=2.3, and that copper is the lowest-resistance metal available. The second, more optimistic scaling will be in line with the SIA roadmap. Certainly, wire performance under the first model will be worse.

3.1 Resistance and capacitance

TABLE 1. Conservative scaling

L_{drawn}	**0.25μm**	**0.18μm**	**0.13μm**	**0.10μm**	**0.07μm**	**0.05μm**
Mid-layer R, Ω/μm	0.07	0.11	0.19	0.32	0.65	1.29
Global layer R, Ω/μm	0.018	0.026	0.044	0.074	0.15	0.30
Mid/Global C, fF/μm	0.23	0.22	0.22	0.20	0.19	0.17

TABLE 2. SIA roadmap scaling

L_{drawn}	**0.25μm**	**0.18μm**	**0.13μm**	**0.10μm**	**0.07μm**	**0.05μm**
Mid-layer R, Ω/μm	0.07	0.11	0.18	0.26	0.39	0.70
Global layer R, Ω/μm	0.018	0.025	0.042	0.061	0.091	0.16
Mid/Global C, fF/μm	0.23	0.19	0.18	0.16	0.16	0.16

In both projections, wire resistance (per unit length) grows under scaling, since the width and height both scale down, despite possible increases in aspect ratio. Capacitance, by contrast, decreases slowly with technology due to projected advances in low-κ dielectrics. We show only one line for capacitance since except for the very top layer, global wires at 16λ pitch have the same capacitance as intermediate layers at 8λ pitch. In the conservative scaling, aspect ratios are capped at around 2 to keep the ratio of sideways to total capacitance under 75%; in the SIA scaling, this ratio is also held under 75%, despite the increase in aspect ratio, because of the low-κ dielectrics that are embedded between metal lines and not between metal layers [11].

3.2 Delay and bandwidth

From the next two tables we see how wire delays, in FO4/mm^2, trend for both scaled and fixed-length or growing wires. For mid-level wires, we show how the delay across the semi-perimeter of a 50K-gate block will grow slowly with conservative scaling, but remain rougly constant with optimistic scaling. The delay of global wires, however, grows aggressively, since a chip edge length increases with technology. The delay of a fixed-length wire (not shown) also grows faster than the scaling factor. Although scaled wire delays do not increase much, they still present problems to design tools and methodologies. A long wire in a 0.25μm technology, if scaled to a 0.07μm technology, does not get

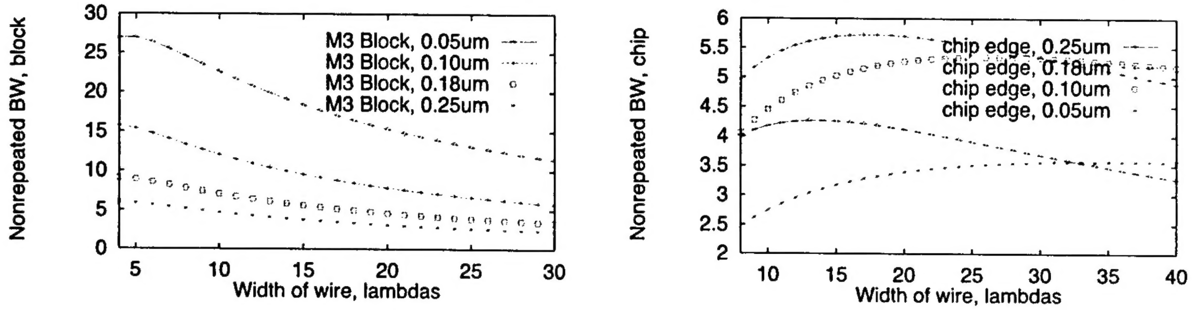
TABLE 3. Delays under scaling

L_{drawn}	**0.25μm**	**0.18μm**	**0.13μm**	**0.10μm**	**0.07μm**	**0.05μm**
50K block semi-perim length, mm	3580	2500	1750	1315	990	740
Conservative 50K block delay, FO4	1.2	1.2	1.4	1.5	2.3	3.4
SIA 50K block delay, FO4	0.94	1.02	1.06	1.03	1.21	1.76
Chip edge, cm	17.3	19	20.7	22.8	24.9	27.4
Conservative chip edge delay, FO4	7.0	15.3	43.7	107	345	1073
SIA chip edge delay, FO4	5.1	13	34	72	180	560

any “longer” (or “shorter”) in terms of delay, but it still needs to be dealt with outside of standard CAD flows. And since design sizes will grow with scaling, the number of these long wires will also increase and put pressure on designer productivity or design team size.

The peak bandwidth of nonrepeated scaled wires is shown below, for both mid-level and global wires. A few generations are omitted for clarity. Mid-level wires span a block of 50,000 gates; global wires span the chip. For brevity’s sake we only show the plots from the SIA scaling; the conservative scaling numbers scale more slowly but show roughly similar trends. We can see that the bandwidth on global wires decreases with scaling after a few generations, but the bandwidth of mid-level wires grows slowly.

FIGURE 7. Unrepeated bandwidth: Mid-level wires on a block and global wires on a chip



3.2.1 Repeaters under scaling

With the delay model and technology scaling assumptions mentioned above, we can predict repeated wire performance for the coming technology generations. All plots show the conservative scaling assumptions on the left and the SIA scaling assumptions on the right.

FIGURE 8. Repeated wire delay (conservative and SIA scaling)

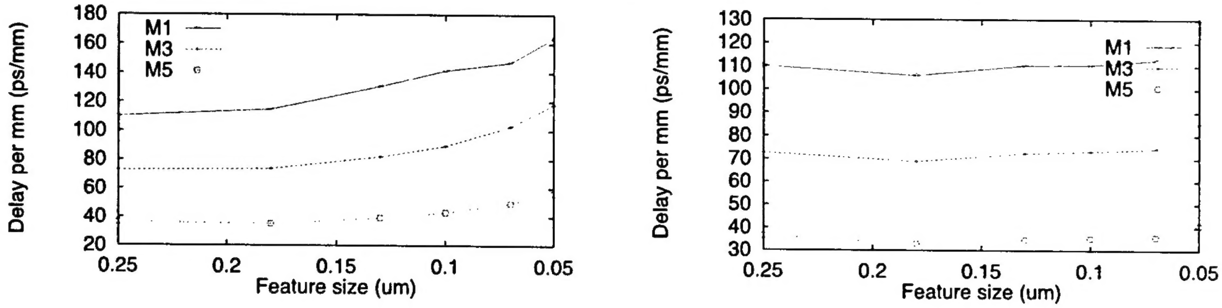
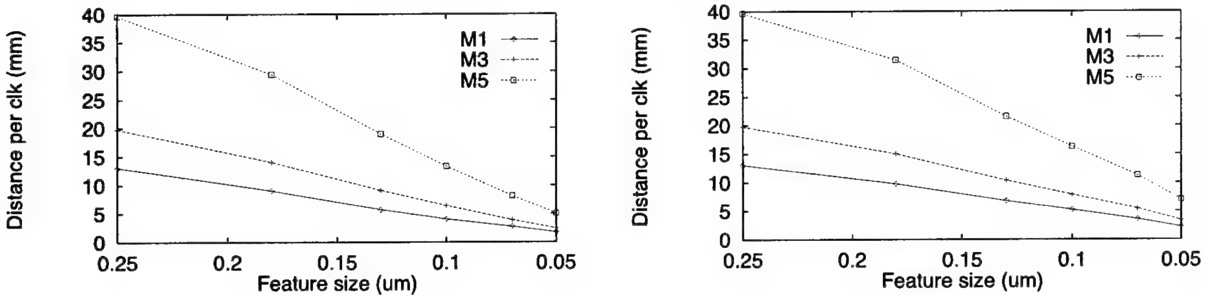


Figure 8 shows that with scaling the delay per unit length (inverse of the propagation velocity) increases modestly for the conservative scaling assumptions and does not change much for the SIA scaling assumptions. For the SIA scaling, the shift to copper wires and low-κ dielectrics accounts for the initial drop in delay between the 0.25μm and 0.18μm technologies. For the successive technology generations, the decreasing resistance per micron of width of the transistors roughly compensates for the increasing wire resistance per unit length, which is also mitigated by better materials and increasing aspect ratios. The increasing wire aspect ratios have the side-effect of increasing the side-to-side capacitance, but improving low-κ dielectrics compensate for this and keep the wire capacitance per unit length roughly constant. The scaling of the gate oxide and device dimensions keeps the transistor gate and diffusion capacitances also roughly constant. The jump in delay at the 0.05μm technology point shows the limits of improving materials technology based on our previous scaling assumptions.

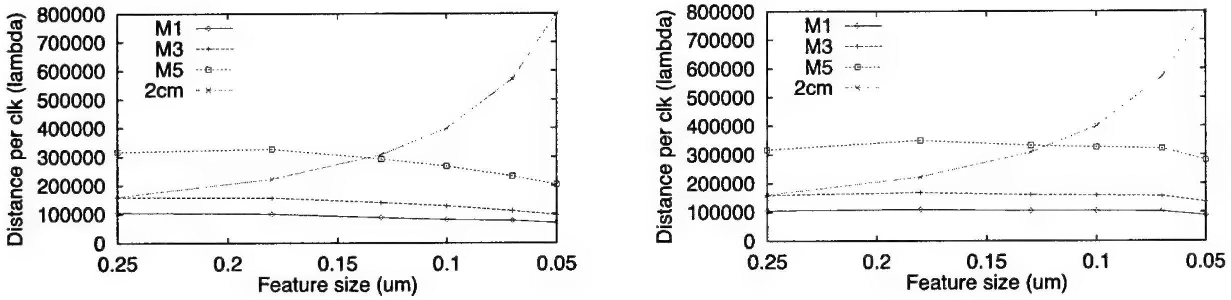
Despite the slowly increasing repeated wire propagation delay, design cycle times will shorten, because the transistors will become faster in successive technology generations. Assuming that clock cycles scale with device speeds, the distance that a signal can travel on a repeated line in a single clock cycle will then decrease. Figure 9 shows the distance a signal can travel in a single clock cycle along a repeated line in various metal layers across technology generations. In Figure 9 and Figure 10, we assume that a future, aggressively clocked design has a clock cycle of 16 FO4.

FIGURE 9. Distance travelled per clock cycle



Although the absolute distance reachable per clock decreases with scaling, the scaled distance reachable per clock changes little in future processes. Figure 10 shows the distance reachable per clock in relative distance units of λs. For the coming technology generations a signal will still be able to travel over 200,000 lambdas in a single clock cycle (even under conservative scaling assumptions) in a repeated upper level metal layer wire. This distance is equivalent to 25mm in a 0.25μm technology.

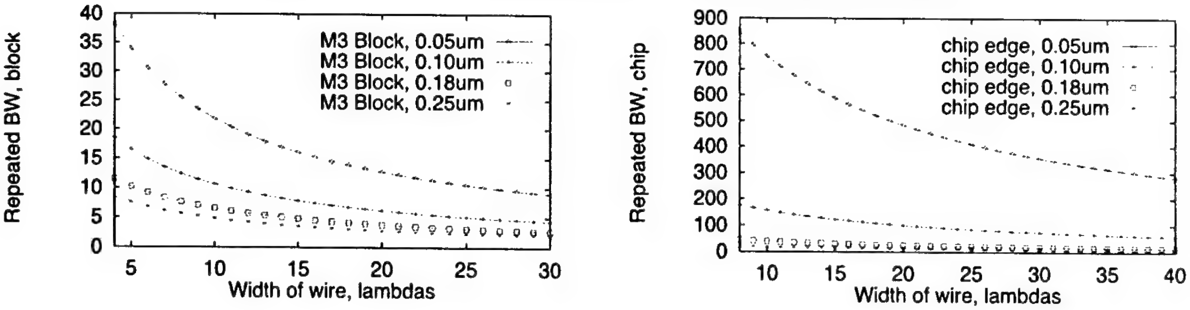
FIGURE 10. Relative distance travelled per clock cycle



Since the scaled distance travelled per clock decreases only slowly, current designs will scale without adverse affects, provided that repeaters are used. However, the decreasing absolute distance reachable per clock is still important, since this implies that today's basic architectures cannot be increased in complexity without encountering adverse interconnect scaling effects, even if repeated lines are used.

We show the bandwidth of repeated lines below. Again, we omit some generations for clarity. Since with repeated lines the bandwidth is independent of wire length we see that it grows rapidly on the global layers with scaling since die sizes grow and allow more wires per chip. The conservative and optimistic scalings make no difference to the repeated bandwidth calculation, since under optimal scaling the segment delay is dependent only on device delays, so we only show one set of graphs.

FIGURE 11. Repeated bandwidth on mid-level and global wires



3.3 Other effects

Coupling effects on both delay and noise mostly depend on the ratio of sideways to total capacitance, and we see that this ratio does not significantly change under scaling. Therefore the delay and noise effects per wire do not dramatically worsen as we advance technologies. However, the number of wires susceptible to these effects will grow with the number of nets on a chip. Since the principal problem today is already one of large design extraction and data management, increased chip sizes of future processes will only exacerbate the problem. Some approaches, such as active noise cancellation [5], can reduce noise but make delay always the worst-case. Other approaches, such as differential signalling, can reduce common-mode noise at the cost of wiring area.

Some recent research, both academic and industrial, has examined on-chip high frequency effects, such as skin effect resistance and inductive effects [12],[13]. The problems in designing for these effects include the chicken-and-egg difficulties in extracting L without first simulating to find where current returns, and also managing and simulating huge netlists with coupled inductors that can span tens or hundreds of microns.

One well-known solution to these problems is to use devoted power planes [14]. The low-resistance and low-inductive return paths on these planes both eliminate large inductive current loops and also make the important skin depth dimension vertical and not horizontal. Thus both wire self-inductance as well as mutual-inductance per unit length is minimized, eliminating so-called slow-mode propagation as well as most coupled noise. Another approach to avoiding skin-effected resistance is to break up wide wires into many interdigitated wires, hence increasing the "skin" available for conduction.

As the number of devices -- and their appetite for current -- on a die grows exponentially under scaling, designers will increasingly find devoted power planes necessary if only to feed enough current to the chip without significant resistive losses. Therefore we predict that these high-frequency effects will remain manageable for the next several generations.

4.0 Machine architecture implications

The previous sections have described an interesting set of constraints for the digital systems architects of the future. In many ways the quality of the wires is not getting much worse. The number of logic gates reachable in a cycle will not change significantly, and the on-chip bandwidth that wires provide will continue to grow. What this picture leaves out is that the number of gates on a chip is growing exponentially, and we are reaching or have reached a point where more gates can fit on a chip than can communicate in one cycle. Said a different way, the absolute distance in mm that a signal can travel in a clock cycle has been decreasing exponentially for a long time, but it never mattered since the numbers were larger than a chip -- until recently. This change has already influenced the design of today's high performance systems, and will have an even larger influence in the future. Luckily, since wires have never been completely free at the board or system level, we know of a number of approaches that can address on-chip communication costs.

The fact that on-chip communication has been cheap for a long time has lead to a number of architectural models that rely on low-latency to shared global resources. These models are attractive to programmers, since they provide the most uniform computational framework and the best functional unit utilization. This focus on function rather than communication is quite pervasive and is the fundamental conceptual roadblock to overcome.

In older technology generations (around 2.0μm processes), the resistance of the wires (even very long ones) was small compared to the resistance of devices. In addition, the density of functional units was low, so it was essential to make maximal use all the functional units. Wire delay was not an issue for these designs -- fitting all the needed functions on-chip was critical.

As technology continued to improve, the resistance of the wires remained small, but the increasing capacitance of the long wires started to be an issue. Floorplanning of the highest-performance designs became important, so the proper sizing could be done to keep communication costs low. For people not needing the maximum performance, wires were still basically ignored.

Continued technology scaling lead to the situation where global wire delays were significant, but still much less than a clock cycle. In this design period, the programming model remained one of global shared resources, but with micro-architectures internally more partitioned. The instruction fetch unit, while logically part of the datapath, started to migrate to the cache location to minimize latency for branch operations. The address adder in many machines was duplicated: one was placed in the datapath where it logically belongs, and a smaller version to generate the cache index was placed near the data cache, again to reduce latency. Wires delays were still modest (much less than a cycle), and the micro-architecture changes were mostly invisible to the user of the device.

Designers developed many tools and methodologies to handle the increasing importance of wires during this period (the beginning of sub-micron design), including analytical models such as wire RC models and now more accurate AWE methods; floorplanning techniques such as delay-driven segregation of local and global routing; and local circuit generation techniques such as layout-driven synthesis. Today's 0.25μm technology designs utilize some or all of the above techniques, since while local routing within reasonably-sized blocks carry wire delays closer to zero, global routes between such blocks are closer to half a cycle. The cost of communication is becoming more explicit. Chips are partitioned early in the design process, and the delay that global lines interconnecting these blocks have are added at this point. One technique is to declare that all block pins have flip-flopped interfaces, thus giving up a clock cycle for communication between blocks. If, after timing analysis, a block has positive slack on its pins, post-placement optimizations of moving logic around between flops may provide some improvement.

As the complexity of the digital systems has continued to increase, architects have responded to higher communication costs by further partitioning the internal micro-architecture and adding internal latency (internal pipe stages) in locations that they think will have the least effect on machine performance. This additional latency allows the machines to absorb the latency cost of the on-chip wires while still taking advantage of the high bandwidths they offer. These added latencies are now visible to the user, but have small effects in the programming model. An example of this effect is in the most recent Alpha processor: the integer unit is partitioned into two clusters, and the latency for communicating between these clusters takes an additional cycle [15].

What will happen as on-chip wire delay begins to take multiple cycles is still an open question. A recent issue of Computer Magazine [16] gives a number of different visions of billion transistor chips and shows the active debate in the computer architecture field about whether one can continue to hide the increasing communication costs in the micro-architecture of the machine. We believe that this will not be possible, and that more explicitly parallel machines will migrate on-chip. This change will in turn lead to interesting new work at exploring computation models that explicitly account for communication costs.

Building machines that have better scaling properties is already an active area of research. These machines are often constructed from processing nodes that don't grow in complexity with technology. Instead, as technology scales, the number of these processing nodes on the chip grows, along with an on-chip communication network. Berkeley has their IRAM and IDisk programs, where they want to build large servers from large number of small processors and disks [17],[18]; MIT has the RAW project, which tries to expose the computation and communication issues to the compiler to see how well it can schedule operations on these machines [19]; and Stanford has the Smart Memory project which is looking at building a flexible collection of processing nodes, memory, and interconnect fabric so it can support a wider variety of programming models efficiently [20]. These are but a few of the research programs in this new area.

In some ways the problems that these future architects face is not that scaled wires are fundamentally bad, it is just our expectations of wires was unreasonable. There is an effective speed of light for on-chip wires, and chip architects need to learn how to deal with it.

5.0 References

[1] Matzke, D. *et al.* "Will physical scalability sabotage performance gains?" *IEEE Computer*, Sept. 1997, p. 37-9.

[2] Rohrer, N. *et al.* "A 480 MHz RISC microprocessor in a 0.12μm L_{eff} CMOS technology with copper interconnects," ISSCC 1998, pp. 240-1

[3] Bohr, M. "Interconnect Scaling - The Real Limiter to High Performance ULSI," IEDM 1995, pp. 241-4.

[4] Bakoglu, H. "Circuits, Interconnections and Packaging for VLSI," Addison-Wesley, 1990.

[5] Naffziger, S. "Design Methodologies for Interconnect in GHZ+ ICs," Tutorial, ISSCC 1999, pp.

[6] Davis, J. *et al.* "A Stochastic Wire-Length Distribution for Gigascale Integration (GSI) -- Part I: Derivation and Validation," IEEE Trans. Electron Devices, vol. 45, No. 3, March 1996, pp. 580-9.

[7] Keckler, S. *et al.* "The MIT Multi-ALU Processor," HotChipsIX 1997, pp. 1-7.

[8] Kuskin, J. *et al.* "The Stanford FLASH Multiprocessor," Proc. 21st Intl. Symp. on Computer Architecture, April 1994, pp. 302-13.

[9] SIA, "National Technology Roadmap for Semiconductors," 1997.

[10] Sylvester, D. and Keutzer, K. "Getting to the Bottom of Deep-Submicron," ICCAD 1998 Tutorial.

[11] Nishi, Y. "The Trend of On-Chip Interconnects: An International Perspective," Presented at Stanford University, 1998 Spring Seminar Series.

[12] Restle, P.J. *et al.* "Designing the Best Clock Distribution Network," Symposium on VLSI Circuits, Dig. Tech. Papers, 1998, pp. 2-6.

[13] Deutsch, A. *et al.* "The importance of inductance and inductive coupling for on-chip wiring," EPEP 1997, pp. 53-6.

[14] Priore, D. "Inductance on silicon for sub-micron CMOS VLSI," Symposium on VLSI Circuits, Dig. Tech. Papers, 1993, pp. 17-18.

[15] Gieseke, B.A. *et al.* "A 600 MHz superscalar RISC microprocessor with out-of-order execution," in *IEEE ISSCC 1997 Dig. Tech. Papers*, Feb. 1997, pp. 176-7.

[16] *IEEE Computer*, Special Issue: Future Microprocessors - How to use a Billion Transistors, September 1997.

[17] Kozyrakis, C.E. *et al.* “Scalable processors in the billion-transistor era: IRAM” *IEEE Computer*, Sept. 1997, pp. 75-8.

[18] Keeton, K. *et al.* “The Intelligent Disk (IDISK): A New Computing Infrastructure for Decision Support Databases,” Presented, National Storage Consortium’s Network Attached Storage Device working group meeting, June 8-9, 1998.

[19] Waingold, E. *et al.* “Baring It All to Software: Raw Machines,” *IEEE Computer*, September 1997, pp. 86-93.

[20] http://velox.stanford.edu/smart_memories